Indag. Mathem., N.S., 10 (1), 145–159 March 29, 1999

Conditional linearization

by R.C. de Vrijer

Department of Theoretical Computer Science, Free University, De Boelelaan 1081a, 1081 HV Amsterdam, the Netherlands.
e-mail: rdv@cs.vu.nl

Communicated by Prof. H.P. Barendregt at the meeting of March 30, 1998

ABSTRACT

A non-leftlinear term rewriting system lacking the Church–Rosser property can sometimes be shown to satisfy the unique normal form property by shifting attention to an associated conditional term rewriting system that is leftlinear. We call this the method of conditional linearization. In the present paper the method is described in a general setting and some applications are discussed. In particular we present a simple proof of the unique normal form property for Combinatory Logic extended with 'Parallel Conditional', that is, with constants C, T and F (conditional, true, false) and extra reduction rules $CTxy \to x$, $CFxy \to y$ and $Czxx \to x$. A special feature of this application is that it involves the use of negative conditions.

INTRODUCTION

A Term Rewriting System (or any Abstract Reduction System for that matter) has the *unique normal form property* (UN), if every convertibility class contains at most one normal form; or equivalently, if convertible normal forms are identical. A TRS satisfying UN is also said to *have unique normal forms.*

In this note we present a simple proof of UN for Combinatory Logic (CL), consisting of the well-known rules for I, K and S, extended with 'Parallel Conditional'. That is, augmented with constants C, T and F (conditional, true, false) and with the extra reduction rules $CTxy \to x$, $CFxy \to y$ and $Czxx \to x$. This TRS, we call it CL-pc, was demonstrated to fail the Church–Rosser property (CR) by Klop [1980]. So the usual way of establishing uniqueness of normal forms, via CR, is not available here.

Our proof for the case of CL-pc is based on a more general method for proving UN for certain term rewriting systems with repeated variables in the left-hand sides of the rules ('non-leftlinear' rules), that has originally been proposed by the present author and was first used in Klop [1980], see also Klop and de Vrijer [1989]. This method proceeds by proving confluence for an associated left-linear *conditional* term rewriting system, that originates from the original non-leftlinear one by – what might be called – 'linearizing' the rules. Apart from the new application to Combinatory Logic with parallel conditional, the purpose of the present paper is to give this method a cogent presentation. The method of conditional linearization is shown to yield very easily yet another interesting result: all TRSs that are non-ambiguous after linearization have unique normal forms (Theorem 3.9).

Two features of the application of the method to Combinatory Logic plus parallel conditional may be worth mentioning. First it involves the use of a Conditional Term Rewriting System with negative conditions, added in order to disambiguate the rewriting rules. Secondly, although the general method is essentially proof-theoretic, our new application uses a lemma that depends on a model-theoretic argument, using the graph model $P\omega$.

The two applications of our method that were mentioned above, also follow from a theorem stated in Chew [1981], establishing uniqueness of normal forms for a wider class of non-leftlinear TRSs. The proof offered by Chew for his theorem seems inconclusive, though. After this became apparent, there has been a renewed interest in finding a complete and convincing proof, notably by Mano and Ogawa [1997]. Chew's theorem will be briefly discussed here in Section 5. We do not go into details of the proofs though.

Anyhow, our proof of UN for CL-pc is independent of Chew's theorem, and thereby much simpler. As a matter of fact, the complexity of both the original proof and the one proposed by Mano and Ogawa surpasses that of the methods used in this note by some orders of magnitude.

Finally, it is worth noting recent related work on the problem of establishing the unique normal form property for non-leftlinear term rewriting systems by Mano, Ogawa, Oyamaguchi, Toyama and Verma. See Section 6.

1. FOUR NON-LEFTLINEAR, NON-CONFLUENT TRSs

In this note we will discuss four specific non-leftlinear extensions of Combinatory Logic: CL-sp, CL-d, CL-e and CL-pc. We recall that CL has a signature consisting of one binary operator, *application*, and three constants, S, K and I. As usual, the application operator is notationally suppressed, its role being taken over by concatenation; we adopt the usual conventions of leaving away brackets, with association to the left. The rewrite rules of CL are:

$$\begin{aligned} \text{CL}: \quad & Sxyz \to xz(yz), \\ & Kxy \to x, \\ & Ix \to x. \end{aligned}$$

The system CL-sp of Combinatory Logic with surjective pairing was the first non-leftlinear term rewriting system to be extensively studied, mostly in the related lambda calculus version (e.g. in Mann [1973], Barendregt [1974], Klop [1980], de Vrijer [1978, 1989], Klop and de Vrijer [1989]). It adds to CL the new constants D, D_1, and D_2, for pairing and its respective projections. The usual rewrite rules are

CL-sp: CL+

$$D_1(Dxy) \to x,$$
$$D_2(Dxy) \to y,$$
$$D(D_1x)(D_2x) \to x.$$

The systems CL-d and CL-e came up in the study of CL-sp; they were proposed by Hindley (see Böhm [1975], Staples [1975]) for theoretical purposes. CL-d adds to CL one new constant D and the non-leftlinear rewrite rule r-d:

CL-d: CL +

r-d: $\quad Dxx \to x.$

In CL-e yet another constant, E, is added. The rule r-e can be seen as test for syntactic identity.

CL-e: CL +

r-e: $\quad Dxx \to E.$

Then finally, the system we are primarily concerned with here augments CL with constants C, T and F, for *conditional*, *true* and *false* respectively. The rewrite rule r-pc makes the conditional *parallel*.

CL-pc: CL +

r-t: $\quad CTxy \to x,$

r-f: $\quad CFxy \to y,$

r-pc: $Czxx \to x.$

Each of these four non-leftlinear rewriting systems lacks the Church–Rosser property (Klop [1980]). But nevertheless, each can be shown to have unique normal forms. Essentially in each of these cases the method of linearizing the rules by adding conditions, described in Section 3 below, can be used. Still, the case of CL-sp is very complicated (see Klop and de Vrijer [1989]) and so is the existing proof of the unique normal form property for CL-pc via Chew's theorem. For the latter case a much simpler proof is presented in this note. The cases of CL-d and CL-e are relatively simple and are included here mainly for expository purposes.

2. CONDITIONAL TERM REWRITING SYSTEMS

A general framework of rewriting that takes the possibility into account that rewrite rules may be subjected to conditions, has probably first been given in

O'Donnell [1977]. Then of course, conditional rewriting has important roots in Universal Algebra and in the field of Algebraic Specifications.

Maybe less well-known, conditional rewriting has yet another origin. Out of the algebraic context, rewriting rules with conditions have been used as a proof-theoretic tool for establishing syntactic properties of unconditional rewriting systems and λ-calculus extensions in Klop [1980], de Vrijer [1987, 1989] and Klop and de Vrijer [1989]. It is the latter kind of use of conditional rewriting that we are concerned with in this note.

Algebraically, conditional rewrite rules can be viewed as implementations of equational specifications containing *positive conditional equations*:

$$(*) \qquad t_1 = s_1 \wedge \ldots \wedge t_n = s_n \Rightarrow t_0 = s_0.$$

If $n = 0$, the equation is unconditional. Conforming with the notation often used in 'equational logic programming', one mostly writes instead of $(*)$:

$$t_0 = s_0 \Leftarrow t_1 = s_1, \ldots, t_n = s_n.$$

Then the transition from conditional equations to conditional rewrite rules can be made by just orienting the equation in the lefthand side. This gives rise to what in Dershowitz, Okada and Sivakumar [1988] has been named *semi-equational* systems. Dershowitz, Okada and Sivakumar list a number of alternative types of CTRSs, thereby extending the classification given in Bergstra and Klop [1986]; the distinctions derive from different choices that can be made in the implementation of the conditions. Apart from the semi-equational systems, we will here make use of one other type of CTRS; it does not correspond to any of the special categories and hence it falls in the inclusive category of *generalized* systems. In generalized systems there is no restriction at all on the character or the format of the conditions; they can be just any predicate.

So we consider the following two types of CTRSs:

(i) *semi-equational* systems

$$t_0 \to s_0 \Leftarrow t_1 = s_1, \ldots, t_n = s_n,$$

(ii) *generalized* systems

$$t_0 \to s_0 \Leftarrow P_1, \ldots, P_n.$$

Note that in the case (i) the definition of $\to$ is circular since it depends on conditions involving a reference to $\to$ (via the conversion relation); but the rewrite rules can be taken as constituting a positive inductive definition of $\to$, since the conditions are positive. In the case of generalized CTRSs one has to take care in formulating conditions involving $\to$, in order to ensure that $\to$ is well-defined.

Note 2.1. Incorporating negative conditions containing $\to$ in a generalized CTRS can be dangerous. Consider the example of CL with constants C and A, and the generalized conditional rule:

$$Cx \to A \Leftarrow \text{ not } x \twoheadrightarrow A.$$

The question is now whether the conditional reduction relation is well-defined. The negative condition 'not $x \twoheadrightarrow A$' is itself in terms of $\to$ and looks circular. Since the condition is negative, the clauses for $\to$ can not, without more, be taken as an inductive definition. Indeed, by a fixed point construction, there is a term Z such that $Z \twoheadrightarrow CZ$. Does $Z \twoheadrightarrow A$ hold? If not, then yes by the conditional rule. If yes, then by which reduction steps?

As a matter of fact, a simpler example already illustrates the point. Consider the generalized CTRS consisting of the single conditional rewrite rule:

$$a \to b \Leftarrow a \neq b.$$

Does $a \to b$ hold?

Note 2.2. A non-leftlinear rule can be seen as a special kind of generalized conditional rewrite rule that is leftlinear. Consider as an illustration the non-leftlinear rule r-d: $Dxx \to x$; in the format of conditional rewriting it becomes

r-d: $\quad Dxy \to x \Leftarrow x \equiv y.$

Recall that an *orthogonal* TRS is one that is unambiguous and left-linear.

Definition 2.3. (i) Let R be a CTRS. Then R_u, the *unconditional version* of R, is the TRS which arises from R by deleting all conditions.

(ii) The CTRS R is called *(non-)leftlinear* if R_u is so; likewise for *orthogonal*.

Definition 2.4. (i) Let R be a CTRS with rewrite relation $\to$, and let P be an n-ary predicate on the set of terms of R. Then P is *stable with respect to* $\to$ if for all terms t_i, t'_i such that $t_i \twoheadrightarrow t'_i$ $(i = 1, \ldots, n)$:

$$P(t_1, \ldots, t_n) \Rightarrow P(t'_1, \ldots, t'_n).$$

(ii) Let R be a CTRS with rewrite relation $\to$. Then R is *stable* if all conditions (appearing in some conditional rewrite rule of R), viewed as predicates with the variables ranging over R-terms, are stable with respect to $\to$.

Theorem 2.5 (O'Donnell [1977]). *Let R be a generalized, orthogonal CTRS which is stable. Then R is confluent.*

The proof is a rather staightforward generalization of the confluence proof for orthogonal TRSs.

Obviously, the convertibility conditions $t_i = s_i$ $(i = 1, \ldots, n)$ in a rewrite rule of a semi-equational CTRS are stable. So the following theorem from Bergstra and Klop [1986] can in fact be obtained as a corollary of Theorem 2.5:

Theorem 2.6. *Orthogonal semi-equational CTRSs are confluent.*

Example 2.7. Let CL-e* be the orthogonal, semi-equational CTRS obtained by extending Combinatory Logic with a 'test for convertibility' (compare CL-e defined in Section 1, with test for syntactic identity):

$$\begin{aligned} & Sxyz \to xz(yz) \\ & Kxy \to x \\ & Ix \to x \\ \text{r-e*:}\quad & Dxy \to E \Leftarrow x = y. \end{aligned}$$

Then R is confluent.

3. APPLICATION OF CTRSs TO PROVE UNIQUENESS OF NORMAL FORMS

In this section we explain the method for proving the property UN for certain non-leftlinear TRSs as a proof-theoretic application of conditional rewriting in the field of term rewriting itself. The method is based on the following simple observation concerning Abstract Reduction Systems (ARSs); recall that an ARS is just any set with a binary relation $\to$, considered as a reduction relation.

Proposition 3.1. *Let R_0 and R_1 be two ARSs with the same set of objects, and with reduction relations $\to_0, \to_1$ and convertibility relations $=_0, =_1$ respectively. Let NF_i be the set of normal forms of R_i $(i = 0, 1)$. Then R_0 is UN if each of the following conditions hold:*
(i) *$\to_1$ extends $\to_0$;*
(ii) *R_1 is CR;*
(iii) *NF_1 contains NF_0.*

Proof. Easy. □

Remark 3.2. The conditions (i) and (ii) could still be weakened to:
(i)′ $=_1$ extends $=_0$;
(ii)′ R_1 is UN.
In the applications that concern us here, however, we use Proposition 3.1 as it is stated. In particular the unique normal form property of R_1 is always obtained as a consequence of confluence.

The interest of Proposition 3.1 derives from its applications, in particular in the method of conditional linearization for proving UN, that is the topic of this paper. By way of illustration, we can already apply it to a relatively simple, but typical example. We consider the non-confluent system CL-e $=$ CL$+$ $\{$r-e : $Dxx \to E\}$ from Section 1. In order to be able to use Proposition 3.1 for establishing UN for CL-e, we 'break' the non-leftlinear constraint in the rule r-e by replacing it with a conditional rule:

$$\text{r-e*:}\qquad Dxy \to E \Leftarrow x = y.$$

Thus we get the system CL-e* of Example 2.7. Remark that the rule r-e* can be seen as resulting from r-e, written in the conditional format of Note 2.2, by relaxing the condition $x \equiv y$ to $x = y$.

Proposition 3.3. *The TRS CL-e has unique normal forms.*

Proof. We want to apply Proposition 3.1 with $R_0 =$ CL-e and $R_1 =$ CL-e*; so we must check the clauses (i), (ii) and (iii).

(i) Obviously $\rightarrow_{\text{CL-e}}$ is contained in $\rightarrow_{\text{CL-e}^*}$, since, as we just observed, the rule r-e: $Dxx \rightarrow E$ can be seen as a restriction of the more liberal conditional rule r-e*: $Dxy \rightarrow E \Leftarrow x = y$. (As a matter of fact, one easily verifies that the convertibility relations of CL-e and CL-e* coincide.)

(ii) The semi-equational CTRS CL-e* is orthogonal; hence, by Theorem 2.6, it is confluent.

(iii) It remains to be checked that each CL-e-normal form t is also a CL-e*-normal form. Consider for a proof by contradiction a term t which is a CL-e-normal form, but not a CL-e*-normal form. Moreover, take t to be of minimal length such that these properties hold. Then t must contain a subterm DXY, such that $X \not\equiv Y$ and $X =_{\text{CL-e}^*} Y$. But then, by the minimality of t, the CL-e-normal forms X and Y must be CL-e*-normal forms as well, convertible but distinct, contradicting the Church–Rosser property of CL-e*. $\square$

In order to make the reasoning of Proposition 3.3 more generally applicable, we introduce the concept of 'linearizing'.

Definition 3.4. (i) If r is a rewritable rule $t \rightarrow s$, we say that $r' = t' \rightarrow s'$ is a *left-linear version* of r if there is a substitution $\sigma : VAR \rightarrow VAR$ such that $r'^{\sigma} = r$ and r' is left-linear.

(ii) If $r = t \rightarrow s$ is a rewrite rule, and $r' = t' \rightarrow s'$ is a left-linear version of r, such that $r = r'^{\sigma}$, then the *conditionalized left-linear version* or *linearization* of r (associated to r') is the conditional rewrite rule:

$$t' \rightarrow s' \Leftarrow \wedge\{x_i = x_j \mid i > j,\ x_i^{\sigma} = x_j^{\sigma},\ x_i,\ x_j \in t'\}.$$

(In case r is already left-linear, it will coincide with its left-linear version r' and with the associated conditional rule.)

Example 3.5. $Czxy \rightarrow y$ is a left-linear version of the non-leftlinear rule $Czxx \rightarrow x$, since using the substitution σ with $\sigma(z) = z$, $\sigma(x) = x$, $\sigma(y) = x$ we have

$$(Czxy \rightarrow y)^{\sigma} = (Czxy)^{\sigma} \rightarrow y^{\sigma} = Czxx \rightarrow x.$$

The associated conditional rule is

$$Czxy \rightarrow y \Leftarrow x = y.$$

Another left-linear version of $Czxx \rightarrow x$ is $Czxy \rightarrow x$, with the associated conditional rule

$$Czxy \to x \Leftarrow x = y.$$

These are the only linearizations, because we will identify rules that originate from each other by a 1-1 renaming of variables as usual.

Definition 3.6 (*Linearization*).

(i) If R is a TRS, then a *linearization* of R is a semi-equational CTRS that consists of linearizations of the rules of R, for each rule of R at least one.

So a linearization R' of R can be obtained by the following two steps:

- Step 1. Choose for every rule $r \in R$ one or more of its left-linear versions; say the (left-linear) rules thus obtained are $r_1, \ldots, r_n$.
- Step 2. Then take R' to be the CTRS consisting of the conditional rewrite rules $r_1^*, \ldots, r_n^*$ associated to $r_1, \ldots, r_n$.

Note that by these two steps the left-linear rules of R are left untouched. In general there will be several linearizations of R, according to the choices that can be made in step 1; but in case R is already left-linear, there is only one, coinciding with R.

(ii) If R is a TRS, then R^L, the *full linearization* of R, is defined as the linearization of R that is obtained by including for each rule $r \in R$ *all* its conditionalized left-linear versions.

Example 3.6. The system CL-e* is the result of linearizing the system CL-e. As a matter of fact, CL-e* = CL-e$^{\mathrm{L}}$.

Lemma 3.7. *Let R' be a linearization of R. Then:*

(i) *The one-step reduction relation of R' extends that of R : $\to_R \subseteq \to_{R'}$.*

(ii) *The conversion relations of R and of R' are the same: $=_R \; = \; =_{R'}$.*

Proof. (i) For each rule r of R at least one of its linearizations r^* is included in R'. In case r itself is left-linear, the rule r coincides with r^*; if r is not left-linear, r is stricter than its linearization r^*.

(ii) The inclusion $=_R \subseteq =_{R'}$ holds because of (i). The inclusion $=_{R'} \subseteq =_R$ follows by induction on conversion in R'. It suffices to check that the rules of R' respect convertibility in R, under the hypothesis that the conditions already hold with respect to R. $\square$

Theorem 3.8. *If a linearization of a term rewriting system R is confluent, then R has unique normal forms.*

Proof. The proof runs parallel to that of Proposition 3.2; it is only a little bit more abstract. So now it suffices to check (i), (ii) and (iii) of Proposition 3.1 for R and a linearization R', that is:

(i) $\to_{R'}$ extends $\to_R$;

(ii) R' is Church–Rosser;

(iii) $NF_{R'}$ contains NF_R.

(i) follows from Lemma 3.7.

(ii) holds by assumption.

As to (iii), we prove by induction on X the implication $X \in NF_R \Rightarrow X \in NF_{R'}$. Assume $X \in NF_R$. Then X can only be not an R'-normal form, if it contains a redex Y that is an instance of a linearization r^* of some non-leftlinear rule $r = t \to s$ of R. That is, $X \equiv C[Y]$ and for a leftlinear version $r' = t' \to s'$ of r (such that $r = r'^{\sigma}$), we have $Y \equiv t'^{\tau}$; moreover the conditions of r^* must be satisfied, amounting to the implication $x_i^\sigma \equiv x_j^\sigma \Rightarrow x_i^\tau = x_j^\tau$, for all $x_i, x_j \in t'$. Since the x_i^τ's are proper subterms of X, and hence R-normal forms, they are by the induction hypothesis also R'-normal forms. Hence, since R' has unique normal forms: $x_i^\sigma \equiv x_j^\sigma \Rightarrow x_i^\tau \equiv x_j^\tau$. But then Y would be also an R-redex, contradicting the assumption that $X \in NF_R$. □

Remark. Like Proposition 3.1, also this theorem may be strengthened by requiring only UN for the linearization.

Now we have obtained a general method to prove uniqueness of normal forms for non-leftlinear TRSs: try to prove CR for one of its linearizations in order to be able to apply Theorem 3.8. Whether the method will work in a particular case, and how difficult it is, depends on the CR problem that ensues.

We first treat a simple but interesting example. Call a TRS *strongly non-ambiguous* if after replacing each non-leftlinear reduction rule by a left-linear version the resulting TRS is non-ambiguous. The following general result is an immediate consequence of Theorem 3.8.

Theorem 3.9. *Any strongly non-ambiguous TRS has unique normal forms.*

Proof. Let R be a strongly non-ambiguous TRS. Consider a linearization R' of R consisting of exactly one conditionalized left-linear version for each rule of R. Then R' will be an orthogonal semi-equational CTRS. Hence the result follows by Theorems 3.8 and 2.6. □

Two examples of non-leftlinear TRSs to which Theorem 3.9 can applied to yield UN are the systems CL-d and CL-e from Section 2. An example of a non-ambiguous but not strongly non-ambiguous TRS that does not have unique normal forms is the following.

Example 3.10 (Huet [1980]). $R = \{F(x,x) \to A, F(x, G(x)) \to B, C \to G(C)\}$. R is non-ambiguous; there are no critical pairs since x and $G(x)$ cannot be unified. However, R is not strongly non-ambiguous, since $\{F(x,y) \to A, F(x, G(y)) \to B\}$ has a critical pair. The term $F(C, C)$ has the two distinct normal forms A and B.

In this section we prove CR for the full linearization CL-pc$^{\mathrm{L}}$ of the ambiguous and non-leftlinear system CL-pc, Combinatory Logic with parallel conditional. Then the uniqueness of normal forms property for CL-pc follows by an application of Theorem 3.8. First we sum up the rules of the linearization CL-pc$^{\mathrm{L}}$.

CL-pc$^{\mathrm{L}}$: CL+

$$
\begin{array}{ll}
\text{r-t:} & CTxy \to x, \\
\text{r-f:} & CFxy \to y, \\
\text{r-pc}^1\text{:} & Czxy \to x \Leftarrow x = y, \\
\text{r-pc}^2\text{:} & Czxy \to y \Leftarrow x = y.
\end{array}
$$

Solving the CR problem may at first look not very promising, because of the vicious cases of overlap between the pairs of rules r-t / r-pc^2, r-f / r-pc^1 and r-pc^1 / r-pc^2. Now the idea is to add extra conditions in order to remove these cases of vicious overlap. This will involve also the use of negative conditions, however, and hence there is the danger of the pitfall indicated in Note 2.1.

To avoid this pitfall we 'fix' the conditions, making them refer to $=_{\text{CL-pc}}$, convertibility in CL-pc. Thereby the conditions have a determinate meaning, independent of the inductive definition of conversion ($=_{\text{CL-pc}^{\text{L-}}}$) they are part of. What we get is not a semi-equational, but a generalized CTRS; it will be called CL-pc$^{\mathrm{L}-}$.

CL-pc$^{\mathrm{L}-}$: CL+

$$
\begin{array}{ll}
\text{r-t:} & CTxy \to x, \\
\text{r-f:} & CFxy \to y, \\
\text{r-pc}^{1-}\text{:} & Czxy \to x \Leftarrow x =_{\text{CL-pc}} y, \text{not } z =_{\text{CL-pc}} F, \\
\text{r-pc}^{2-}\text{:} & Czxy \to y \Leftarrow x =_{\text{CL-pc}} y,\ z =_{\text{CL-pc}} F.
\end{array}
$$

Lemma 4.1.

(i) *The convertibility relations in CL-pc, in CL-pc^{L}, and in CL-pc^{L-} coincide.*

(ii) $\to_{\text{CL-pc}^{\text{L-}}} \subseteq \to_{\text{CL-pc}^{\text{L}}}$.

Proof. (i) For conversion in CL-pc and CL-pc$^{\mathrm{L}}$ we have Lemma 3.7. We show that $=_{\text{CL-pc}^{\text{L-}}} = =_{\text{CL-pc}}$. The inclusion $=_{\text{CL-pc}} \subseteq =_{\text{CL-pc}^{\text{L-}}}$ holds since each instance of the rule r-pc of CL-pc is also an instance of either r-pc^1 or r-pc^2. So it suffices to check that the rules r-pc^1 and r-pc^2 of CL-pc$^{\mathrm{L}-}$ respect convertibility in CL-pc. This is immediate by the (positive) conditions $x =_{\text{CL-pc}} y$.

(ii) By (i), the conditions $x =_{\text{CL-pc}} y$ of the rules r-pc^{1-} and r-pc^{2-} amount to the same as the conditions on r-pc^{1,2}. Then the extra conditions on r-pc^{1,2} can only make the relation $\to_{\text{CL-pc}^{\text{L-}}}$ stricter than $\to_{\text{CL-pc}^{\text{L}}}$. □

Now in order to prove CR for CL-pc$^{\mathrm{L}-}$ we need to know that $T \neq_{\text{CL-pc}} F$; this will guarantee that there is indeed no overlap in CL-pc^{L-} between the rules r-t and r-pc^2, etc. A model construction within the Graph Model $P\omega$ for CL can be used for this purpose.

The Graph Model $P\omega$ is surveyed e.g. in Barendregt [1981], Chapter 18; we assume the following preliminaries.

- The function $(\ ,\ ) : \omega \times \omega \to \omega$ is a 1-1 coding of pairs of natural numbers;
- $e(0), e(1), e(2), \ldots$ is a list of all finite subsets of ω, with $e(0)$ the empty set;
- the function $s : \omega \to \omega$ is such that $e(s(n)) = \{n\}$, for all $n \in \omega$ (s stands for 'singleton').

Lemma 4.2. $T \neq_{\text{CL-pc}} F$

Proof. The following definitions of T, F and C within $P\omega$ can be given, satisfying the equations of CL-pc:

$$\begin{aligned} T &= \{1\}, \\ F &= \{0\}, \\ C &= \{(0,(s(n),(s(n),n))) \mid n \in \omega\} \cup \{(s(1),(s(n),(0,n))) \mid n \in \omega\} \cup \\ &\qquad \{(s(0),(0,(s(n),n))) \mid n \in \omega\}. \end{aligned}$$

The model thus obtained satisfies $=_{\text{CL-pc}}$, but not $T = F$. □

Proposition 4.3. *The system CL-pc^{L-} is Church–Rosser.*

Proof. By Lemma 4.1 it follows that the conditions of CL-pc^{L-} are stable. Moreover between the rules of CL-pc^{L-} there are no harmful cases of overlap, due to the negative condition and to Lemma 4.2. Then proving CR is a routine matter (compare Theorems 2.5 and 2.6). □

The confluence of CL-pc^{L} can now be concluded from Lemma 4.1 and Proposition 4.3 by using the following general principle for ARSs.

Proposition 4.4. *Let R and R' be ARSs such that the following three conditions are satisfied:*

(i) *R' is confluent*

(ii) $\to_{R'} \subseteq \twoheadrightarrow_R$

(iii) $=_R \subseteq =_{R'}$

Then R is confluent.

Proof. Assume $t =_R s$. Then by clause (iii) also $t =_{R'} s$. Hence by (i), the terms t and s must have a common reduct in R'. But then, by (ii), t and s have the same common reduct in R. □

Theorem 4.5.

(i) *The system CL-pc^{L} is Church–Rosser.*

(ii) *The system CL-pc has unique normal forms.*

Proof. (i) The Church–Rosser property for CL-pc$^{\mathrm{L}}$ follows by Proposition 4.4 from Proposition 4.3 and Lemma 4.1.

(ii) Since we have confluence for the linearization CL-pc$^{\mathrm{L}}$, Theorem 3.8 can be applied. □

5. CHEW'S THEOREM

We will now give a brief account of a theorem stated in Chew [1981], giving sufficient conditions for a TRS to have unique normal forms. In the light of the present paper, Chew's theorem can be viewed as a generalization of Theorem 3.9: the condition of strong non-ambiguity is relaxed to allow overlap at the root between the lefthand sides of rules, but only when an extra requirement is met, called *compatibility* (see Definition 5.1). The paradigmatic example of a non-leftlinear TRS that is not strongly non-ambiguous, but still within the scope of Chew's theorem, is the system CL-pc.

So Chew's conditions imply both our Theorems 3.9 and 4.5(ii). The proof in Chew [1981], however is far more complicated than the ones given here. For the relation between our method involving the use of an associated CTRS and Chew's approach see the remarks below.

Definition 5.1 *(Compatibility)*.

(i) Let $r = t \to s$ be a rewrite rule. Then the set of all left-linear versions of r,

$$\{t' \to s'_1, \ldots, t' \to s'_n\}$$

is a *cluster* of rewrite rules. (Note that the left-hand sides of the rules in the cluster corresponding to r, are taken the same. In Chew [1981] this cluster is presented as $t' \to \{s'_1, \ldots, s'_n\}$.)

(ii) Now let $r_1 : t_1 \to s_1$ and $r_2 : t_2 \to s_2$ be two different rewrite rules of the TRS R. Let

$$\{t'_1 \to s'_{1i} \mid i = 1, \ldots, n\} \text{ and}$$
$$\{t'_2 \to s'_{2j} \mid j = 1, \ldots, m\}$$

be the two clusters corresponding to r_1 and r_2, respectively. We say that R has *compatible* rewrite rules (or that R is compatible) if for all r_1, r_2 the following holds:

(a) t'_1 cannot be unified with a proper subterm of itself. Likewise for t'_2.

(b) t'_1 cannot be unified with a proper subterm of t'_2. Likewise with 1, 2 interchanged.

(c) if t'_1, t'_2 can be unified (at the root), via mgu σ, then the two clusters must have a common σ-instance:

$$\{(t'_1 \to s'_{1i})^\sigma \mid i = 1, \ldots, n\} \cap \{(t'_2 \to s'_{2j})^\sigma \mid j = 1, \ldots, m\} \neq \emptyset.$$

Note 5.2. The terminology used here deviates slightly from that in Chew [1981]. There the notion *strongly non-overlapping* allows possible overlap at the root; the term *compatible* only concerns condition (ii)(c) of Definition 5.1.

Theorem 5.3 (Chew [1981]). *Let R be a compatible TRS; then R is UN.*

Like the method we described in Section 3, the proof of Theorem 5.3 given in Chew [1981] does rely on Proposition 3.1. Also analogously, the extended rewriting relation (R_1) used by Chew is the result of some procedure of linearizing the non-leftlinear rules. But Chew does not make use of the notion of conditional rewriting, and accordingly his linearizations are slightly different from the ones obtained via an associated semi-equational CTRS. Then the Church–Rosser proof by Chew for the linearizations he obtains from compatible systems is by an ingenious and complex syntactic analysis.

It seems not unlikely that Chew's approach can be transferred to our CTRS framework. If R is a TRS and R^L its full linearization, we can group the conditional rules of R^L also in clusters, according to how they originated from rules in R. Thus for example for CL-pc, we have the partition in clusters (indicated by boxes) in Figure 5.1.

$$\boxed{Sxyz \to xz(yz)}$$

$$\boxed{Kxy \to x}$$

$$\boxed{Ix \to x}$$

$$\boxed{CTxy \to x}$$

$$\boxed{CFxy \to y}$$

$$\boxed{\begin{array}{l} Czxy \to x \Leftarrow x = y \\ Czxy \to y \Leftarrow x = y \end{array}}$$

Figure 5.1

Now it should be proved that the full linearizations of compatible TRSs satisfy CR. Then it would follow by Theorem 3.8 that all compatible TRSs are UN.

Conjecture 5.4. Let R be a compatible TRS and let the semi-equational CTRS R^L be the full linearization of R. Then R^L is confluent. (Hence R is UN.)

6. REMARKS AND FURTHER QUESTIONS

6.0. The origin of this article is the unpublished report de Vrijer [1990]. In a condensed form the results were presented in Klop and de Vrijer [1991]. Since then there has been a renewed interest in the subject of uniqueness of normal forms for non-leftlinear systems. In Toyama and Oyamaguchi [1994] and Mano and Ogawa [1996] the method of conditional linearization has been extended or adapted to specific situations. Verma [1997] proposes different methods with the aim of arriving at similar results.

6.1. As said before, the original proof of Theorem 5.3 is by an ingenious but also very complicated syntactic analysis. In de Vrijer [1990] we reported that we had not succeeded in fully reconstructing all details of the argument from the rather sketchy presentation in Chew [1981]. While visiting NTT Basic Research Laboratories in 1994, van Oostrom explicitly pointed out a gap in Chew's proof (see Mano and Ogawa [1997]). This inspired Mano and Ogawa to invest a considerable effort in finding a new proof. The resulting proof is presented in Mano and Ogawa [1997]. It is more perspicuous but certainly no less complicated than Chew's proof.

6.2. Combinatory Logic with parallel conditional is presented here and in Chew [1981] as the paradigmatic example of a compatible, not strongly non-ambiuous TRS. We do not know yet another interesting example. It would be interesting to know if such examples exist. A related question is whether it would be possible to broaden the scope of Theorem 5.3 by extending Chew's syntactic analysis beyond the class of compatible TRSs.

6.3. There do exist non-leftlinear systems known to have unique normal forms that are not compatible and are therefore outside the scope of Chew's theorem. An example of such a TRS is CL-sp. It is covered in Klop and de Vrijer [1989].

6.4. It is at present an open question whether another linearization of CL-pc, the system CL-pc^1, is confluent. This question was suggested in a personal communication by Toyama.

CL-pc^1: CL+

$$CTxy \to x,$$
$$CFxy \to y,$$
$$Czxy \to x \Leftarrow x = y.$$

ACKNOWLEDGEMENTS

I would like to thank Yoshihito Toyama for some interesting communications via e-mail on the subject of this paper, and Jan Willem Klop, Aart Middeldorp and Vincent van Oostrom for their support.

REFERENCES

Barendregt, H.P. – Pairing without conventional restraints. Zeitschrift für Math. Logik und Grundl. der Math. **20**, 289–306 (1974).

Barendregt, H.P. – The Lambda Calcalus, its Syntax and Semantics. 1st ed. North-Holland 1981, 2nd ed. North-Holland 1984 (1981).

Bergstra, J.A. and J.W. Klop – Conditional rewrite rules: confluence and termination. Journal of Computer and Systems Sciences **32**, 323–362 (1986).

Böhm, C. – λ-Calculus and Computer Science Theory. Springer Lecture Notes in Computer Science **37**, (ed.) (1975).

Chew, P. – Unique normal forms in term rewriting systems with repeated variables. In: 13th Annual ACM Symposium on the theory of Computing, 7–18 (1981).

Dershowitz, N., M. Okada and G. Sivakumar – Canonical Conditional Rewrite Systems. In: Proc. of 9th Conf. on Automated Deduction, Argonne. Springer LNCS **310**, 538–549 (1988).

Huet, G. – Confluent reductions: Abstract properties and applications to term rewriting systems. Journal of the ACM **27**, No. 4, 797–821 (1980).

Klop, J.W. – Combinatory Reduction Systems. Mathematical Centre Tracts **127**, Centre for Mathematics and Computer Science, Amsterdam (1980).

Klop, J.W. and R.C. de Vrijer – Unique normal forms for lambda calculus with surjective pairing. Information and Computation **80**, 97–113 (1989).

Klop, J.W. and R.C. de Vrijer – Extended Term Rewriting systems. In: Conditional and Typed Rewriting Systems. Proc. of the 2nd International CTRS Workshop, Montreal 1990 (eds. S. Kaplan and M. Okada), Springer Lecture Notes in Computer Science **516**, 26–50 (1991).

Mano, K. and M. Ogawa – Unique normal form property of higher-order rewriting systems. In: Proc. of the 5th International Conference on Algebra and Logic Programming. Springer Lecture Notes in Computer Science **1139**, 269–283 (1997).

Mano, K. and M. Ogawa – A new proof of Chew's theorem. Preprint, NTT Basic Research Laboratories (1997).

Mann, C.R. – Connections between proof theory and category theory. Dissertation, Oxford University (1973).

O'Donnell, M.J. – Computing in systems described by equations. Springer Lecture Notes in Computer Science **58** (1977).

Ogawa, M. – Chew's theorem revisitied - uniquely normalizing property of non-linear term rewriting systems. In: Proc. of the International Symposium on algorithms and computation. Springer Lecture Notes in Computer Science **650**, 309–318 (1992).

Staples, J. – Church–Rosser theorems for replacement systems. In: Algebra and Logic (ed. J.N. Crossley). Springer Lecture Notes in Mathematics **450**, 291–307 (1975).

Toyama, Y. and M. Oyamaguchi – Church–Rosser property and unique normal form property of non-duplicating term rewriting systems. In: Proc. of the 4th International CTRS. Springer Lecture Notes in Computer Science **968**, 316–331 (1995).

Verma, R.M. – Unique normal forms for nonlinear Term Rewriting Systems: root overlaps. In: Fundamentals of computation theory. Proc. of the 11th International Symposium FTC, Kraków, 1997 (eds. B.S. Chlebus and L. Czaja). Springer Lecture Notes in Computer Science **1279**, 452–462 (1997).

Vrijer, R.C. de – Surjective pairing and strong normalization: two themes in lambda calculus. Dissertation, University of Amsterdam (1987).

Vrijer, R.C. de – Extending the lambda calculus with surjective pairing is conservative. In: Proc. of the 4th Annual IEEE Symposium on Logic in Computer Science, 204–215 (1989).